

Individueel plan Gert

Als individueel onderdeel van ons project ga ik een aantal heuristieken uitwerken. Deze heuristieken worden een method van onze Algorithm class. Een groot deel van deze heuristieken past de score van de kwaliteitsfunctie aan. Ook daarvoor zal ik de methods schrijven.

Uiteindelijke functionaliteiten

Zonder de pretentie maar met de ambitie alles te (kunnen) schrijven, hebben we een lijst met heuristieken:

- zgn. 'eindstations' (stations met nog één connectie over) definiëren, en nieuwe trajecten hiermee laten beginnen
- een maximaal aantal keer dat een connectie in een route mag zitten.
- een maximaal aantal keer dat een trein heen en weer kan gaan tussen twee stations
- connecties die zijn 'ingesloten' verbieden/minpunten geven. Een ingesloten connectie heeft alleen maar bezette connecties om zich heen.
- minpunten geven voor het aantal minuten dat een route onder het time_frame zit, oftewel: aantal connecties in een route maximaliseren
- zgn. 'moeilijke' connecties die bereden zijn pluspunten geven. Moeilijke connecties zijn connecties tussen twee drieslagen.

Te schrijven methods

- Method(s) voor het berekenen van plus- en minpunten op de score van de state:
 - Method die plus- en minpunten van alle gebruikte heuristieken bij elkaar optelt
 - Method die de score van een state muteert met verkregen plus- en minpunten
- Methods voor de eindstations heuristiek:
 - Method die een lijst geeft met eindstations
 - Method die definieert of een station een eindstation is
- Methods voor max aantal keer heen en weer gaan:
 - Method die bekijkt hoe vaak een connectie vlak voor het keuzepunt al bereden is
 - Method die alle connecties van een keuzepunt geeft die niet verboden zijn.
 - Method die minpunten geeft voor elke connectie die meer dan de max bereden is.
- Methods voor maximaliseren aantal connecties in route:
 - Method die aantal minpunten bepaalt voor een route
 - Method die aantal minpunten bepaalt voor alle routes
- Methods voor pluspunten moeilijke connecties:
 - Method die bepaalt of een connectie moeilijk is.
 - Method die een lijst geeft met alle moeilijke connecties
 - Method die pluspunten geeft voor alle bereden moeilijke connecties

Aansluiting met werk van andere studenten

We hebben al een Algorithm class geïmplementeerd. Alle andere algoritmes zullen deze class inheriten. In deze class zal ik al deze methods schrijven, die dan dus voor ieder ander algoritme beschikbaar zijn.

Wat anders uit is gepakt

Er zijn geen onverwachte dingen gebeurd. Eigenlijk liep alles volgens plan. Het enige wat een beetje anders was, is dat ik soms een extra method nodig had voor het implementeren van een heuristiek. Dat zijn de volgende methods:

- De method die kijkt hoe vaak een connectie herhaaldelijk bereden is voor een keuzepunt, moest door de inrichting van onze datastructuur worden opgesplitst in twee methods, één voor het eerste station en één voor het laatste station. Dat heb ik opgelost door één uniforme method te maken, die wordt aangeroepen door de twee bovenstaande methods. Dit om herhaling van code te voorkomen.
- Voor de method die plus- en minpunten geeft voor het herhaaldelijk berijden van connecties, heb ik een method geschreven die dit doet voor één specifieke connectie.

Links naar commits

1. [Individueel plan gemaakt](#)
2. [Individueel plan aangepast](#)
3. [Individueel plan afgemaakt](#)
4. [Eerste heuristiek toegevoegd](#)
5. [Heuristieken voor herhaling van connectiegebruik toegevoegd](#)
6. [Heuristiek voor routemaximalisatie toegevoegd](#)
7. [Heuristiek voor moeilijke connecties toegevoegd](#)
8. [Enkele vergeten regels code toegevoegd](#)